NAVAL POSTGRADUATE SCHOOL
MONTEREY, CALIFORNIA

THESIS

INTERACTIVE TOOLS FOR SOUND SIGNAL ANALYSIS/SYNTHESIS BASED ON A SINUSOIDAL REPRESENTATION

by

Ming-Fei Chuang

March 1997

Thesis Advisor:	Charles W. Therrien
Second Reader:	Roberto Cristi

Approved for public release; distribution is unlimited.

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
March 1997

3. REPORT TYPE AND DATES COVERED
Master's Thesis

4. TITLE AND SUBTITLE INTERACTIVE TOOLS FOR SOUND SIGNAL ANALYSIS/SYNTHESIS BASED ON A SINUSOIDAL REPRESENTATION

5. FUNDING NUMBERS

6. AUTHOR(S) Ming-Fei Chuang

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey, CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(maximum 200 words)*

This thesis develops a series of programs that implement the sinusoidal representation model for speech and sound waveform analysis and synthesis. This sinusoidal representation model can also be used for a variety of sound signal transformations such as time-scale modification and frequency scaling. The above sound analysis/synthesis sinusoidal representations and transformations were developed as two interactive tools with Graphical User Interface (GUI) using MATLAB. In addition, an interactive tool for signal frequency component editing based on the sinusoidal model is also presented in this thesis.

14. SUBJECT TERMS Sinusoidal Representation, Analysis/Synthesis, GUI, STFT, Frequency Track, Speech, Time-Scale Modification, Frequency Scaling

15. NUMBER OF PAGES 56

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT
Unclassified

18. SECURITY CLASSIFICATION OF THIS PAGE
Unclassified

19. SECURITY CLASSIFICATION OF ABSTRACT
Unclassified

20. LIMITATION OF ABSTRACT
UL

INTERACTIVE TOOLS FOR SOUND SIGNAL ANALYSIS/SYNTHESIS BASED ON A SINUSOIDAL REPRESENTATION

Ming-Fei Chuang
Lieutenant, Republic of China Navy
B.S., Chung-Chang Institute of Technology, 1992

Submitted in partial fulfillment
of the requirements for the degree of

**MASTER OF SCIENCE
IN
ENGINEERING ACOUSTICS**

from the

**NAVAL POSTGRADUATE SCHOOL
March 1997**

ABSTRACT

This thesis develops a series of programs that implement the sinusoidal representation model for speech and sound waveform analysis and synthesis. This sinusoidal representation model can also be used for a variety of sound signal transformations such as time-scale modification and frequency scaling. The above sound analysis/synthesis sinusoidal representations and transformations were developed as two interactive tools with Graphical User Interface (GUI) using MATLAB. In addition, an interactive tool for signal frequency component editing based on the sinusoidal model is also presented in this thesis.

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGEMENTS

Thanks first go to my thesis advisor, Professor Charles W. Therrien. He is always patient in answering any question that I might have. This thesis could not be completed without his excellent instruction. Thanks also to my good friends, Steve Bergman, Hakki Celebioglu, Natanael Ruiz, James Scrofani, and Charles Victory, their good friendship and sense of humor have made my two-year study experience much more endurable. Last, but not the least, I would like to thank my parents for their continuous encouragement and support.

I. INTRODUCTION

A. OVERVIEW

Sinusoidal representation is a useful model for speech and sound analysis/synthesis. It has been shown that the synthetic waveform preserves the general waveform shape and is perceptually indistinguishable from the original sound [Refs. 1,2,3]. However, in a number of applications it is required to transform a sound signal to a different waveform which is more useful than the original. For example, in time-scale modification of speech, the rate of articulation may be slowed down in order to make degraded speech more comprehensible. Alternatively, the sound can be speeded up so we can quickly scan a passage or compress it into a fixed time interval. In other applications, the sound is compressed or expanded in time or frequency. For instance, in music synthesis it is useful to change the length or pitch of a tone without changing its tonal quality or timbre. In all of these cases, it is desired to perform sound modification. This thesis implements a fixed time-scale modification and frequency scaling based on the sinusoidal model [Ref. 4]. The above sound analysis/synthesis sinusoidal representations and transformations have been developed as two interactive tools with Graphical User Interface (GUI) using MATLAB.

Since some frequency components in a signal are redundant (they may either correspond to the noise, or carry no information), we may not want to include them when we resynthesize the sound signal. In other cases, only a portion of the original signal needs to be regenerated, or a small part of the signal is required to be repeated at a specific time instant. In these applications, we need to have the ability to edit the frequency components of the synthetic signal. Thus, an interactive tool for signal editing based on the sinusoidal model is also presented in this thesis.

B. THESIS OUTLINE

The remainder of this thesis is organized as follows. Chapter II addresses the sine-wave speech model in two parts. First, the speech analysis/synthesis model based on a sinusoidal representation is presented [Ref. 1]. Following this, algorithms for fixed time-scale modification and frequency scaling based on the sinusoidal representation are introduced [Ref. 4]. Chapter III describes the implementation of the three interactive tools for sound analysis/synthesis with GUI. The methods implemented include signal editing, fixed time-scale modification, and frequency scaling. Results of their use are also shown here. Finally, Chapter IV gives conclusions and recommendations for future work.

II. ANALYSIS / SYNTHESIS BASED ON A SINUSOIDAL REPRESENTATION

A. SINE-WAVE ANALYSIS / SYNTHESIS MODEL

1. The Sinusoidal Representation

A speech modeling technique developed by McAulay and Quatieri [Ref. 1], is based upon a sinusoidal representation of the original waveform. In the general speech production model, the output waveform is assumed to be the result of passing the glottal excitation $e(t)$ through a linear time-varying filter $h(t,\tau)$ which models the vocal tract. The model can be written as

$$s(t) = \int_0^t e(\tau)\, h(t, t-\tau)\, d\tau \ . \tag{1}$$

and is depicted in Figure 1.

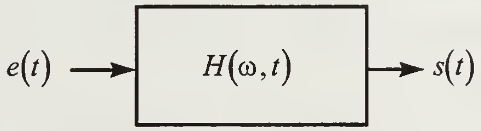


Figure 1. Model of Speech Production

In the sinusoidal model the excitation is written as a sum of sinusoids with time-varying amplitudes and phases, namely

$$e(t) = \sum_{\ell=1}^{L(t)} a_\ell(t) \cos\left[\Omega_\ell(t)\right] \ , \tag{2}$$

where $\Omega_\ell(t)$ is given by

$$\Omega_\ell(t) = \Psi_\ell(t) + \phi_\ell \ , \tag{3}$$

with

$$\Psi_\ell(t) = \int_{t_\ell}^{t} \omega_\ell(\sigma)\, d\sigma \ , \tag{4}$$

In this model, t_ℓ is the onset time of the ℓ^{th} sine wave and $L(t)$ is the number of sine-wave components at time t. For the ℓ^{th} sine-wave component, $a_\ell(t)$ is the time-varying amplitude, $\omega_\ell(t)$ is the frequency and $\Omega_\ell(t)$ the phase corresponding to the ℓ^{th} sine wave. The quantity $\Psi_\ell(t)$ is the time-varying contribution to the phase while ϕ_ℓ is a fixed phase offset needed since the sine wave components for different indices ℓ are generally not aligned.

If the vocal tract transfer function is written as

$$H(\omega,t) = M(\omega,t)\exp\left[j\Phi(\omega,t)\right] , \tag{5}$$

then the output speech $s(t)$ can be written as

$$s(t) = \sum_{\ell=1}^{L(t)} A_\ell(t)\cos\left[\theta_\ell(t)\right] , \tag{6}$$

where

$$A_\ell(t) = a_\ell(t)M_\ell(t) , \tag{7}$$

and

$$\theta_\ell(t) = \Omega_\ell(t) + \Phi_\ell(t) , \tag{8}$$

are the amplitude and phase of the ℓ^{th} sine wave component corresponding to the frequency $\omega_\ell(t)$.

The sine wave model has been found to be useful for modeling other types of sounds besides speech. Other specific applications for this method have been in music synthesis [Ref. 1, 2] and in underwater acoustics [Ref. 3]. For these applications, the separation of the sound into excitation and system components as shown in Figure 1 may or may not be appropriate. Still, most of the basic ingredients of the model remain and can be applied in these applications.

2. Analysis

The purpose of the analysis step is to estimate the composite amplitudes, frequencies, and phases of the sine wave model. This can be done from the high-resolution short-time Fourier transform (STFT). The original analysis method proposed by McAuley and Quatieri [Ref. 1] uses a purely sine-wave-based model (i.e., the excitation and system contributions of each sine-wave component are not explicitly represented). In their following work, a new analysis procedure is developed which separates the vocal cord excitation and vocal tract system contributions as described above. Since we are interested in more general types of sounds rather than speech, the original analysis method along with the modified amplitude and phase representations is used. Thus, we account only for the model of the vocal cord excitation contribution and ignore the vocal tract system contribution. With this simplification, Eq. (7) and Eq. (8) become

$$A_\ell(t) = a_\ell(t), \tag{9}$$

and

$$\theta_\ell(t) = \Omega_\ell(t) \ . \tag{10}$$

The analysis proceeds as follows. First, the data is sectioned into frames of equal length for spectral analysis and a Hamming window is applied before taking the Fourier transform. Frames are formed at an interval of less than the frame length allowing for overlap of data. For a speech signal, a frame length of $20 \sim 30$ milliseconds and overlap interval of $10 \sim 15$ milliseconds are recommended [Ref. 1]. If the Fourier transform of the windowed speech segment is written as $S(\omega, kR)$, then the frequencies of $e(t)$ in Eq. (2) at time kR (i.e., the k^{th} analysis frame), are chosen to correspond to the $L(kR)$ largest peaks in the magnitude of the short-time Fourier transform, $|S(\omega, kR)|$. The locations of the largest peaks are estimated by looking for a change of slope from positive to negative of the Fourier transform magnitude.

If we denote the frequency estimate of the ℓ^{th} sinusoidal component at the k^{th} analysis frame by $\hat{\omega}_\ell^k = \hat{\omega}_\ell(\text{kR})$, then the amplitudes and phases of the sine-wave component are given by the samples of $S(\omega, kR)$ at the specific frequency positions. In other words, the amplitudes and phases are written as

$$\hat{a}_\ell^k = \left|S\left(\hat{\omega}_\ell^k, kR\right)\right| \;, \tag{11}$$

and

$$\hat{\Omega}_\ell^k = \arg\left[S\left(\hat{\omega}_\ell^k, kR\right)\right] \;, \tag{12}$$

where "arg" denotes the principal phase value. A block diagram of the analysis scheme is given in Figure 2.

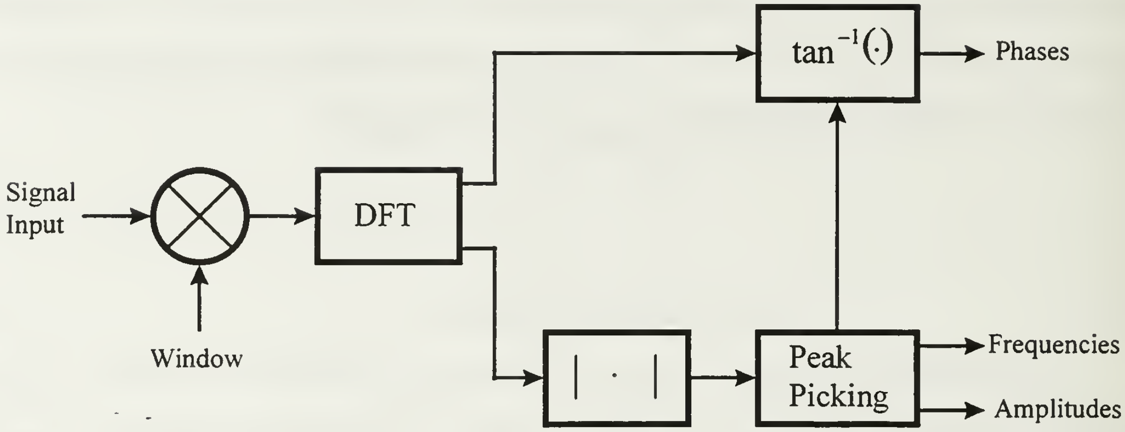


Figure 2. Block Diagram of Sinusoidal Analysis

The number of peaks are not constant from frame to frame in general, and there will be spurious peaks due to the effects of window sidelobe interaction. In addition, the locations of the peaks will change as the pitch changes; and rapid changes in both the location and number of peaks often occur in certain regions of the sound signal. In order to account for such movements in the spectral peaks, the concept of "birth" and "death" of sinusoidal components is introduced here. Suppose that the peaks up to frame k have been matched and a new parameter set for frame k + 1 is generated. We now attempt to match frequency ω_n^k in frame k to the frequencies in frame $k + 1$. If all frequencies ω_m^{k+1} in frame $k + 1$ lie outside a "matching interval" of ω_n^k, then the frequency track associated with ω_n^k is declared "dead" on entering frame k + 1. When all frequencies of frame k have been tested and assigned to continuing tracks or to dying tracks, there may remain frequencies in frame $k + 1$ for which no matches have been made. It is assumed that such frequencies ω_m^{k+1} were "born" in frame k and a new frequency ω_m^k is created in frame k with zero magnitude.

This procedure is done for all unmatched frequencies. Further details of this "birth" and "death" matching procedure can be found in [Ref. 1].

The result of applying this method to a segment of a sound signal is shown in Figure 3. Each horizontal line represents a particular frequency component which is present for some number of frames. These lines are called "frequency tracks." The frequency tracks demonstrate the ability of the method to adapt quickly through the transitory regions such as voiced/unvoiced transitions in speech. Typically there are many very short frequency tracks. Some of these may not contribute significantly to the general structure of the waveform but merely serve to match small details. As will be seen later, the editing tools developed in this thesis allow one to eliminate many of these shorter frequency tracks and thus simplify the sinusoidal model for the signal.

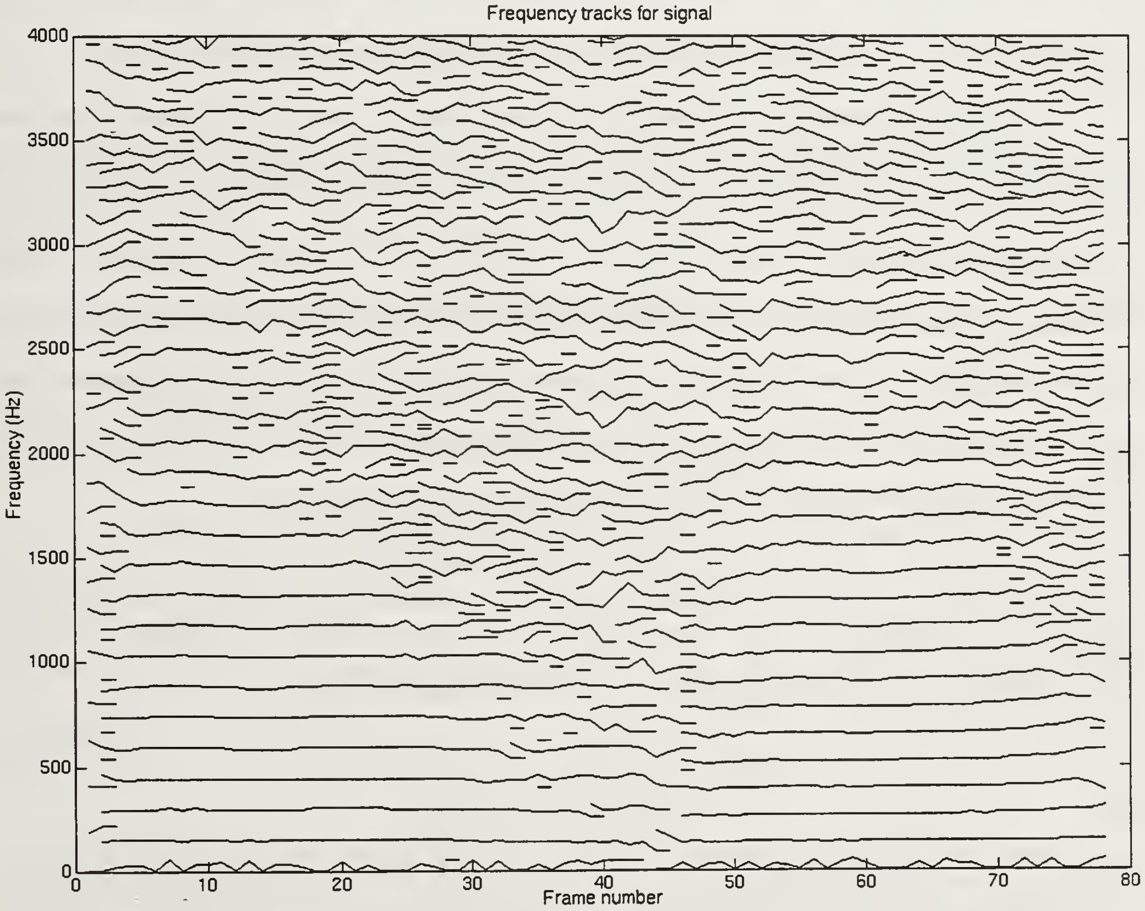


Figure 3. Typical Frequency Tracks for a Sound Signal

3. Synthesis

Sound signal synthesis from the sine-wave parameters begins with matching the amplitude and phase samples in Eq. (11) and Eq. (12) of each sine-wave computed at consecutive frame boundaries. This is followed by interpolation of the resulting pairs of amplitude and phase samples of the signal over each frame. The interpolation of parameters is based on the assumption that the signal is "slowly varying" across each frame and that the frequencies of the sine waves form smooth frequency tracks $\omega_\ell(t)$. This constraint allows us to interpolate samples over a frame duration. If linear interpolation is used for the amplitude, the amplitude estimate $\hat{a}_\ell(t)$ over the k^{th} frame is given by

$$\hat{a}_\ell(t) = \hat{a}_\ell^k + \left(\hat{a}_\ell^{k+1} - \hat{a}_\ell^k\right) t\big/T \ , \qquad \textbf{(13)}$$

where $\hat{a}_\ell^k$ and $\hat{a}_\ell^{k+1}$ are a successive pair of excitation amplitude estimates for the ℓ^{th} frequency track, T is the frame duration and $t \in [0,T]$ is the time into the k^{th} frame.

This simple linear interpolating procedure cannot be used for estimating the phase and frequency of the sinusoid over a frame, however. This is because the phase $\hat{\Omega}_\ell^k$ may contain discontinuities of 2π since the phase of $S(\omega, kR)$ in Eq. (12) is measured modulo 2π. Hence, phase unwrapping must be performed for interpolation of the excitation phase to ensure that the frequency tracks are sufficiently "smooth" across the frame boundaries. A cubic polynomial for solving this problem is first proposed in [Ref. 1] for sine-wave-based synthesis. For the duration at a *single* frame the estimate is defined as

$$\hat{\Omega}_\ell(t) = a + bt + ct^2 + dt^3 \ , \qquad \textbf{(14)}$$

with $t = 0$ corresponding to frame k and $t = T$ corresponding to frame $k+1$. The instantaneous frequency is then the derivative of the phase, namely

$$\hat{\omega}_\ell(t) = \frac{d}{dt}\hat{\Omega}_\ell(t) = b + 2ct + 3dt^2 \ . \qquad \textbf{(15)}$$

In order to provide a good synthetic waveform, it is necessary that the cubic phase function and its derivative equal the excitation phase and frequencies measured at the frame boundaries. By using the algorithms in [Ref. 1], the resulting phase function not only matches the phase at the frame boundaries, but also resolves the 2π phase discontinuities. Details of phase unwrapping and cubic interpolation can be found in [Ref. 1].

It was noted earlier that the phase estimate over the k^{th} frame can be written in terms of a time-varying term and a constant. Specifically, from Eq. (3) and Eq. (4)

$$\begin{aligned}\hat{\Omega}_\ell(t) &= \int_{t_\ell}^{t} \hat{\omega}_\ell(\sigma)d\sigma + \hat{\phi}_\ell \\ &= \int_{t_\ell}^{0} \hat{\omega}_\ell(\sigma)d\sigma + \int_{0}^{t} \hat{\omega}_\ell(\sigma)d\sigma + \hat{\phi}_\ell \ ,\end{aligned} \tag{16}$$

where the time origin ($t\ =\ 0$) is taken to be at the beginning of the current frame and t_ℓ is the onset time of the ℓ^{th} sine-wave. Let $\sum_\ell^k$ denote the phase due to the time-varying frequency accumulated up to frame k; that is,

$$\sum\nolimits_\ell^k = \int_{t_\ell}^{0} \hat{\omega}_\ell(\sigma)d\sigma \ . \tag{17}$$

If $\hat{V}_\ell(t)$ denotes the phase due to the time-varying frequency accumulated over frame k; that is,

$$\hat{V}_\ell(t) = \int_{0}^{t} \hat{\omega}_\ell(\sigma)d\sigma \ , \tag{18}$$

then the excitation phase can be written as

$$\hat{\Omega}_\ell(t) = \hat{V}_\ell(t) + \sum\nolimits_\ell^k + \hat{\phi}_\ell \ . \tag{19}$$

The resulting excitation phase function $\hat{\Omega}_\ell(t)$ consists of a constant component and a time-varying portion. The constant component consists of two parts: the phase offset estimate $\hat{\phi}_\ell$, and the accumulated phase component $\sum_\ell^k$, which can be obtained recursively as

$$\sum\nolimits_\ell^{k+1} = \sum\nolimits_\ell^k + \hat{V}_\ell(T) \ . \tag{20}$$

The interpolated amplitudes and phases are used to generate sinusoids which are then summed to generate the output sound signal. The final synthetic waveform is written as

$$\hat{s}(t) = \sum_{\ell=1}^{L(t)} \hat{a}_\ell(t) \cos\left[\hat{\Omega}_\ell(t)\right] \tag{21}$$

where

$$\hat{\Omega}_\ell(t) = \hat{V}_\ell(t) + \sum\nolimits_\ell^k + \hat{\phi}_\ell \ . \tag{22}$$

A block diagram of the synthesis structure is given in Figure 4.

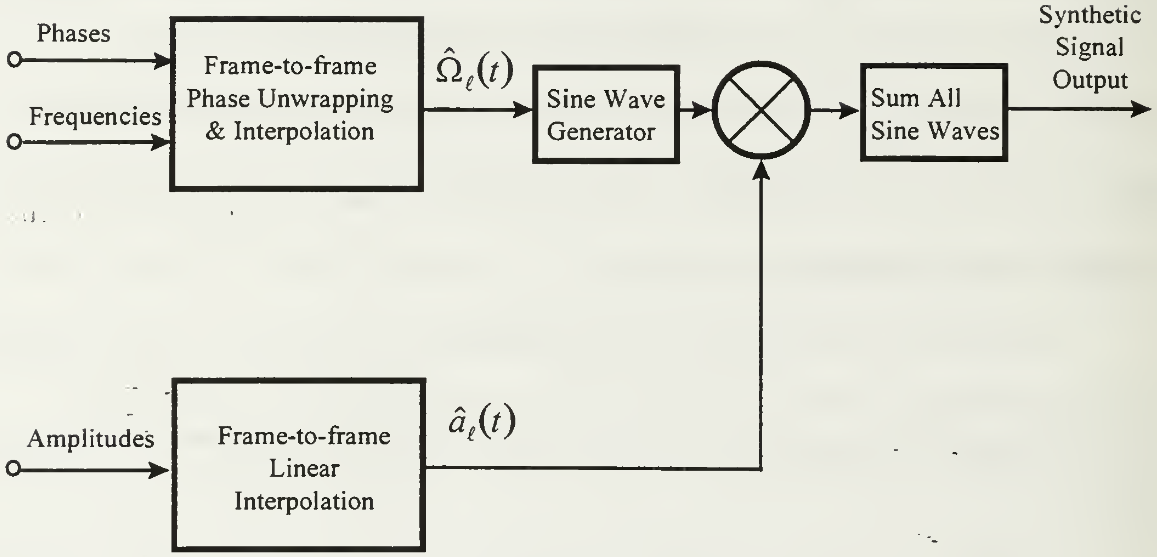


Figure 4. Block Diagram of Sinusoidal Synthesis

B. TIME-SCALE AND FREQUENCY TRANSFORMATION

1. Fixed Rate Change

The goal of time-scale modification is to maintain the perceptual quality of the original sound while changing the apparent rate of sound production. In speech, the technique is used to synthesize speech corresponding to a person speaking more rapidly or slowly without changing the quality of the person's voice. The scheme illustrated here is based upon the algorithm developed by Quatieri and McAulay with slight simplification [Ref. 4]. Although the authors proposed both fixed rate change and time-varying rate change, only the fixed rate change is performed here.

For a fixed time-scale transformation, the time t_0 corresponding to the original sound production rate is mapped to the transformed time t_0' through the mapping $t_0' = \rho t_0$. The case $\rho > 1$ corresponds to time-scale expansion, while the case $\rho < 1$ corresponds to time- scale compression. The case of time-scale expansion is depicted in Figure 5.

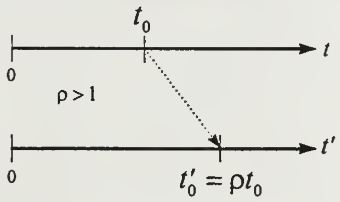


Figure 5. Time Warping with Fixed Rate Change $\rho > 1$

In the sine-wave model discussed here, the parameters which are scaled are the model amplitudes, frequencies, and phases. The model parameters are modified so that frequency tracks $\omega_\ell(t)$ are stretched or compressed in time while the value of $\omega_\ell(t)$, which corresponds to pitch, is maintained. The mathematical model for the fixed time-scale modified sound $s'(t')$, is then given by

$$s'(t') = \sum_{\ell=1}^{L(t')} a_\ell'(t') \cos\left[\Omega_\ell'(t')\right] , \tag{23}$$

where

$$a_\ell'(t') = a_\ell(\rho^{-1}t') , \tag{24}$$

and

$$\Omega_\ell'(t') = \int_{t_\ell'}^{t'} \omega_\ell(\rho^{-1}\tau) d\tau + \phi_\ell . \tag{25}$$

Letting $\sigma = \rho^{-1}\tau$, then $\Omega_\ell'(t')$ can be written as

$$\begin{aligned} \Omega_\ell'(t') &= \int_{t_\ell}^{\rho^{-1}t'} \omega_\ell(\sigma) d\sigma / \rho^{-1} + \phi_\ell \\ &= V_\ell(\rho^{-1}t')/\rho^{-1} + \left(\sum_\ell^k \ \right)' + \phi_\ell . \end{aligned} \tag{26}$$

Since these model parameters are derived on a frame-by-frame basis, we can think of the inverted time $\rho^{-1}t'$ as the time into the k^{th} frame within the original time scale. Therefore, the fixed time-scale synthetic waveform can be obtained as

$$\hat{s}'(t') = \sum_{\ell=1}^{L(t')} \hat{a}'_\ell(t') \cos\left[\hat{\Omega}'_\ell(t')\right] , \tag{27}$$

where

$$\hat{a}'_\ell(t') = \hat{a}_\ell\left[\left(\rho^{-1}t'\right)_T\right] \quad , \tag{28}$$

and

$$\hat{\Omega}'_\ell(t') = \hat{V}_\ell\left[\left(\rho^{-1}t'\right)_T\right]/\rho^{-1} + \left(\Sigma_\ell^k\right)' + \hat{\phi}_\ell \ , \tag{29}$$

and $\left(\Sigma_\ell^k\right)'$ computed recursively as

$$\left(\Sigma_\ell^{k+1}\right)' = \left(\Sigma_\ell^k\right)' + \hat{V}_\ell(T)/\rho^{-1} \ . \tag{30}$$

The notation $(\ )_T$ in Eq. (28) denotes modulo T, which is the original frame duration. Figure 6 illustrates an example in which a segment of male speech is expanded by a factor of 2.

2. Frequency Scaling

The sound can be changed in pitch by performing frequency scaling. This is accomplished by taking the synthesized phase to be

$$\begin{aligned} \Omega'_\ell(t) &= \int_{t_\ell}^{t} \beta\omega_\ell(\tau)d\tau + \phi_\ell \\ &= \beta V_\ell(t) + \phi_\ell \ , \end{aligned} \tag{31}$$

where β is the scaling factor for each frequency track $\omega_\ell(t)$. The operation performed here is equivalent to shifting the frequency tracks to new locations. The resulting modified waveform over the k^{th} frame is given by

$$\hat{s}'(t) = \sum_{\ell=1}^{L(t)} \hat{a}_\ell(t) \cos\left[\hat{\Omega}'_\ell(t)\right] , \tag{32}$$

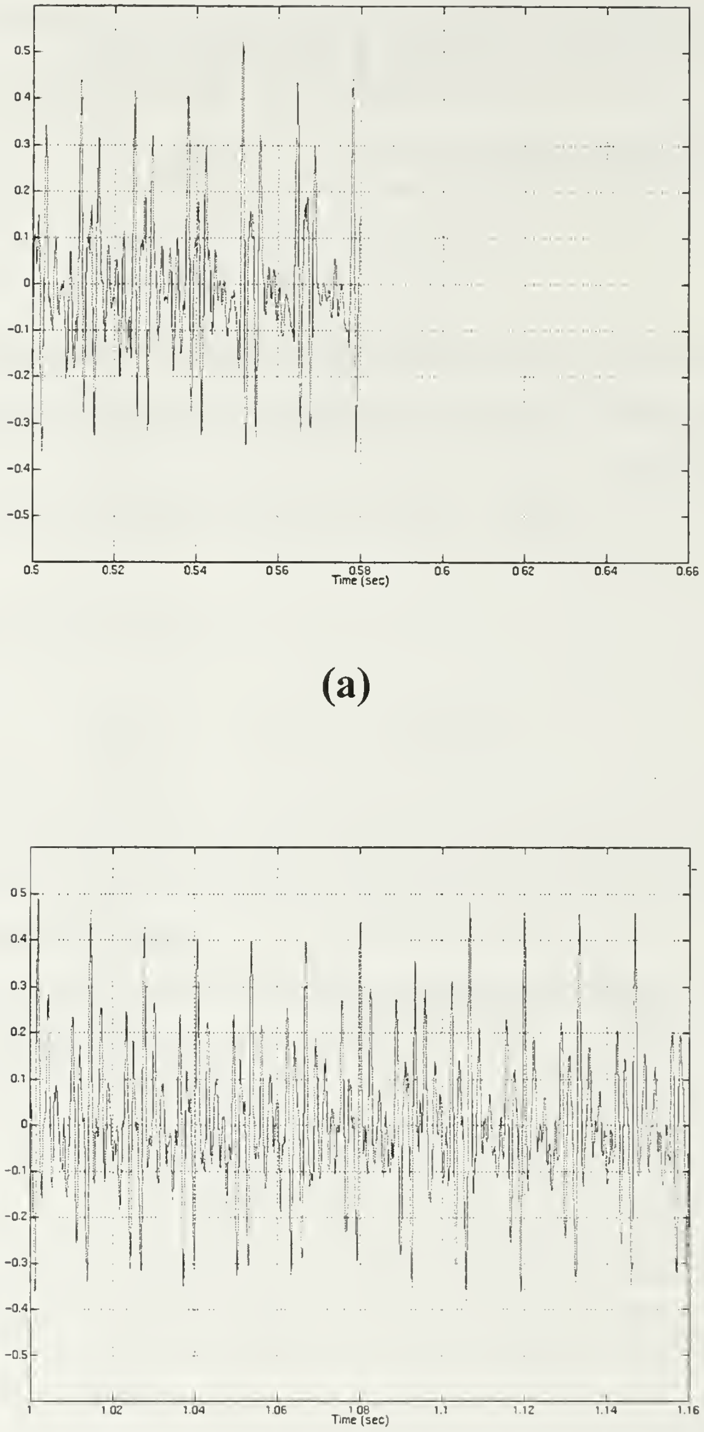
where

$$\hat{\Omega}'_\ell(t) = \beta \hat{V}_\ell(t) + \left(\sum_\ell^k\right)' + \hat{\phi}_\ell \ , \tag{33}$$

with $\left(\sum_\ell^k\right)'$ computed recursively as

$$\left(\sum_\ell^{k+1}\right)' = \left(\sum_\ell^k\right)' + \beta \hat{V}_\ell(T) \ . \tag{34}$$

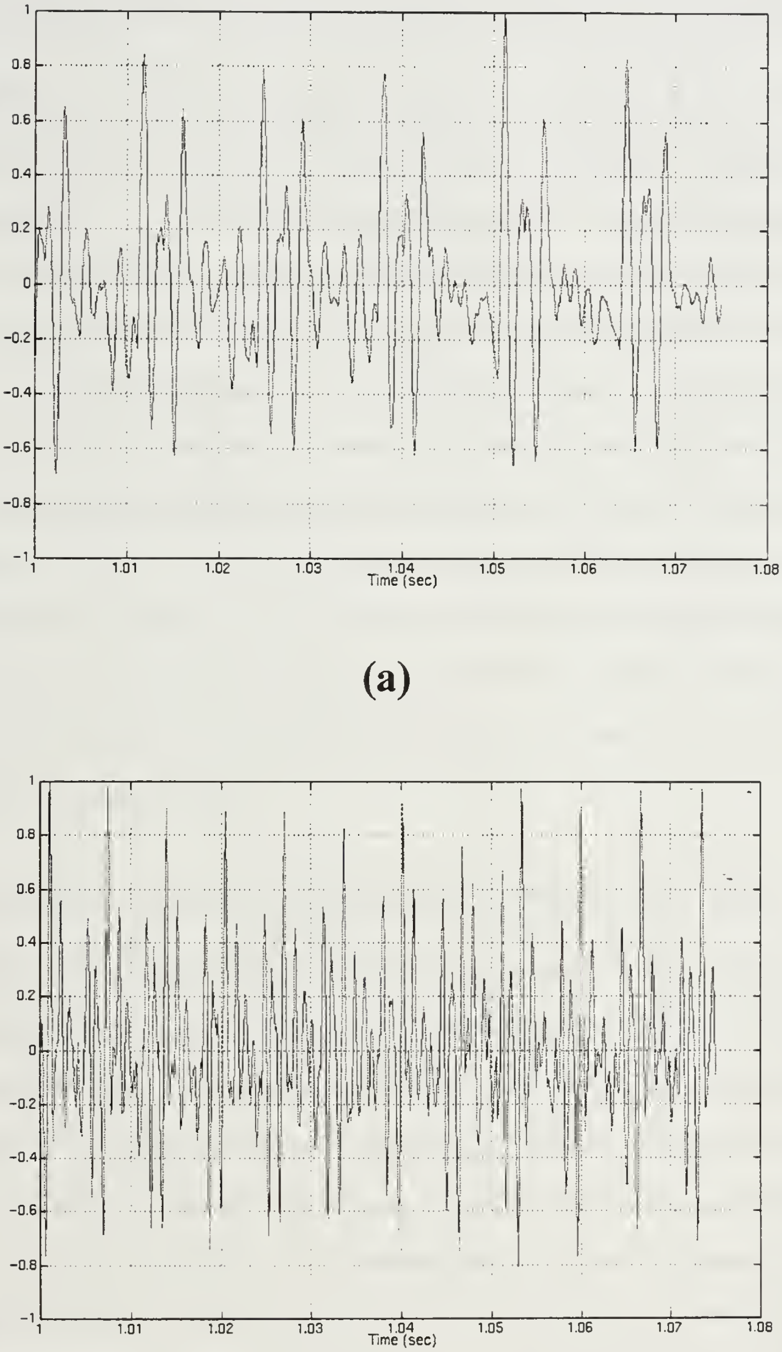
This waveform modification corresponds to an expansion or compression of frequency and a change in pitch. Figure 7 illustrates an example in which the pitch of a male speech is scaled by a factor of 2.



(a)

(b)

Figure 6. Time-scale Expansion of Speech (a) Original (b) Expansion (ρ=2)



(a)

(b)

Figure 7. Frequency Scaling of Speech (a) Original (b) Pitch-scaled (β=2)

III. IMPLEMENTATION OF THE INTERACTIVE TOOLS

The definition of a user interface is moving from a command-line oriented interface to one that includes graphic features. Over the years, graphical user interfaces (GUI) have grown in popularity. GUI use push buttons, editable boxes, and other graphical controls which can be activated with a mouse to select various options and execute commands [Ref. 5]. The purpose here was to develop interactive user interface tools which can be used to perform the sound analysis/synthesis, frequency track editing, and sound transformations based on the sinusoidal representation. The use of these GUI relieves the user of the need to memorize a large number of textual commands, and allows him/her to see the results almost immediately. The interactive tools described here were developed on Unix workstations, and require MATLAB version 4.2c as well as its Signal Processing Toolbox. Although MATLAB provides the necessary support for the GUI on both Unix and IBM PC-compatible platforms, some modifications will need to be made if the user wants to use these tools on IBM compatible PCs.

A. THE SOUND ANALYSIS / SYNTHESIS INTERACTIVE TOOL

An interactive sound analysis/synthesis tool based on sinusoidal representation model is described in this section. This tool allows the user to analyze an existing sound waveform, extract the parameters that represent a quasi-stationary portion of that waveform, and then use those parameters to reconstruct an approximation that is “very close” to the original signal. In other words, the algorithms behind this tool contain two parts, namely, analysis and synthesis. When this tool is invoked, the user must indicate a sound signal as an input argument in the associated .m function; the signal will be drawn in the top portion of the window as shown in Figure 8 (a) and labeled “Original Signal.” After loading the signal into the workspace, the user needs to provide some important values which are used in the signal analysis and synthesis algorithms.

The first value is the sampling frequency for the analysis and synthesis procedures which should be same as the value used in digitizing the original sound signal. For all cases discussed in this thesis, the sampling frequency 8,000 Hz is used. This is close to the actual value of 8,192 Hz used in the SUN Unix workstations.

The next values to be entered are the windowed frame length and overlap width. A windowed frame greater than 20 milliseconds is sufficient for generating a good quality synthetic waveform according to [Ref. 1]; this corresponds to 160 points if the sampling frequency is 8,000 Hz. A 50% overlap of the frame is recommended for this sinusoidal representation model, which would result in a frame overlap width of 80 points. The default values for the windowed frame length and overlap width in this tool are 200 and 100 points, respectively. These two user-input values are shown in the second and third editable boxes in Figure 8 (a) and (b).

The fourth parameter is the threshold level (in dB), which allows the user to limit the maximum number of peaks detected over a frame. The typical range for this threshold value is from 60 to 90 dB. A default value of 80 dB has been used throughout the experiments. In general, the performance will not be affected much by the choice of this threshold level unless too few peaks are allowed.

A concept of "birth" and "death" of sinusoidal components was described earlier in Chapter I (B) and is used to account for the rapid change on both the number and location of spectral peaks. The fifth input value in this tool is the frequency interval used while the frame-to-frame peak matching procedure is performed. It indicates the number of frequency bins that the frequencies on two successive frames can deviate and still be considered to be "matched." A value of 10 has been set as a default.

The last input value is the number of points used in the computation for discrete Fourier transform (DFT) of each frame. Typically, 512 to 1024 points should be enough for generating the synthesis signal if the frame length does not exceed 500 points. In this thesis, all experiments were done using 1024 points.

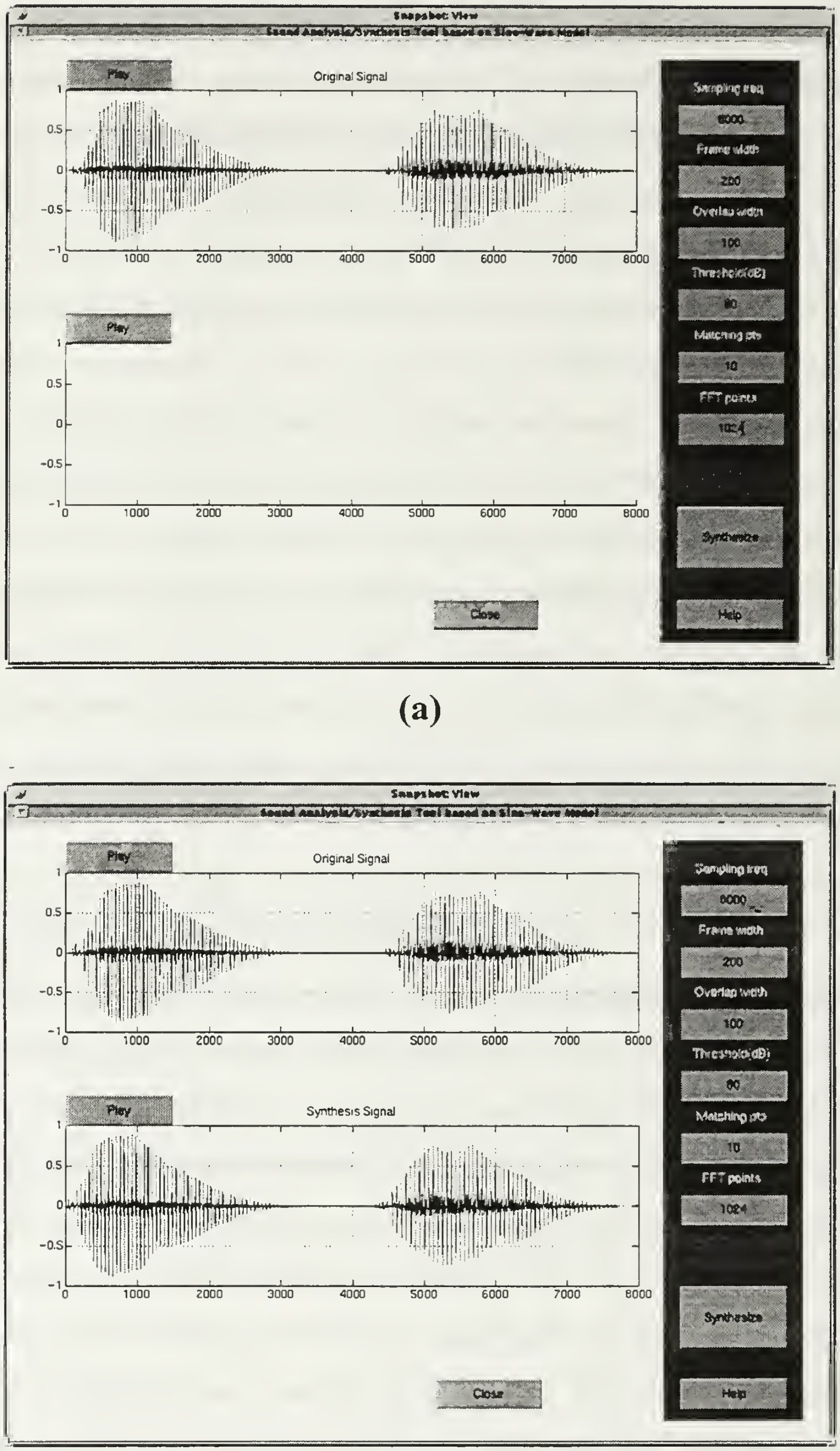
Having entered these values the user now is ready to do the sound signal analysis/synthesis. The "Synthesize" push button activates both the analysis and the synthesis functions. In other words, when the user presses the "Synthesize" button, the tool extracts the waveform parameters first, which are required for the model, and then passes those parameters to the synthesis system so that the synthetic waveform will be generated. The result after entering the parameters and pressing the "Synthesize" button is as shown on the bottom portion in Figure 8 (b).

Of the above six user-input values, only the first three values (i.e., sampling frequency, windowed frame length, and overlap width) are essential and case-dependent for the sound signal analysis/synthesis. It is usually not necessary to change the other three values (i.e., threshold level, frequency matching interval, and DFT points). Additionally, push buttons are available for the users to "play" (i.e., listen to) both the original and synthesis sound signal using the platform's audio output. For new users and users that are not familiar with this interactive tool, an on-line help function is available by pressing the "Help" push button.

B. THE FREQUENCY TRACK EDITING INTERACTIVE TOOL

The frequency track editing tool allows the user to "edit" frequency components of a signal by inputting some appropriate parameters and even by using a pointing device such as a mouse. In a number of applications, it is desired to make the synthetic signal fit in a specific time interval, or to eliminate short frequency tracks to simplify the model. Frequently, some of these short tracks only serve to match the "detail" in the original waveform, and removing them will not change the general characteristics of the sound.

Since the results of the sinusoidal representation model are very robust, it is not necessary to reconstruct the signal with all frequency components which are extracted from the original waveform. This interactive tool offers five frequency track editing functions for users who wish to generate different types of synthetic signals.



Snapshot: View

Sound Analysis/Synthesis Tool based on Sine-Wave Model

Sampling freq: 8000

Frame width: 200

Overlap width: 100

Threshold(dB): 80

Matching pts: 10

FFT points: 1024

(a)

Snapshot: View

Sound Analysis/Synthesis Tool based on Sine-Wave Model

Sampling freq: 8000

Frame width: 200

Overlap width: 100

Threshold(dB): 80

Matching pts: 10

FFT points: 1024

(b)

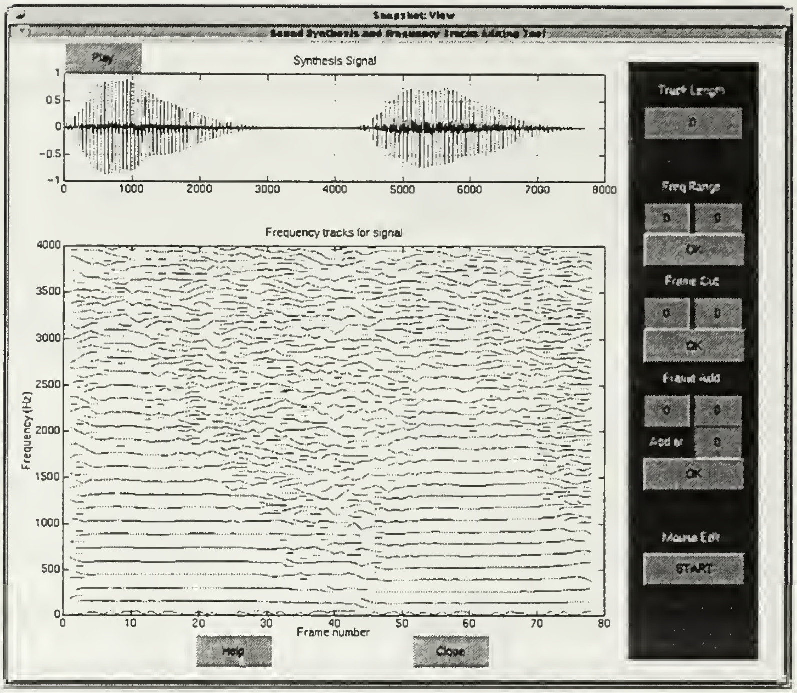
Figure 8. Sound Analysis/Synthesis Interactive Tool (a) Before Synthesis (b) After Synthesis

The first option provided by this tool allows the user to eliminate all frequency tracks of less than some specified length. After eliminating these tracks, the associated signal is resynthesized according to the new frequency tracks. Figure 9 illustrates an example in which all frequency tracks less than 20 frames in length are removed. Notice that there is no large difference between the "original" synthesis waveform in (a) and the "new" synthesis waveform in (b), although the underlying model has been considerably simplified.

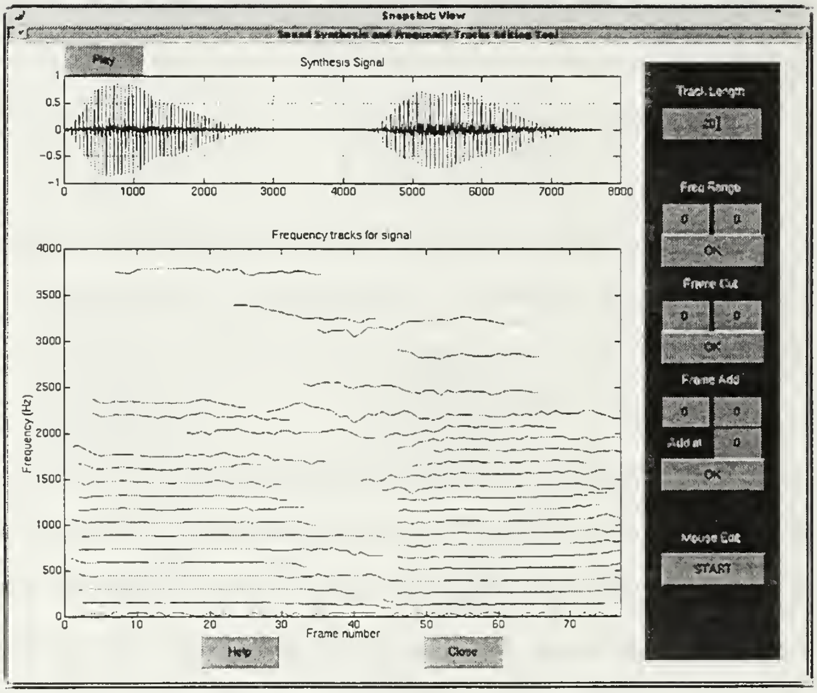
In many signal processing applications, users are likely to design different kind of filters, such as low pass, high pass, or band pass filters, in order to eliminate the unwanted frequency components and preserve the specific range of frequencies which carries the information they need. The second option offered by this tool allows the user to implement those filters very easily, just by indicating the specific range of frequencies to be removed. The example in Figure 10 illustrates the result of eliminating frequencies in the range from 3,000 Hz to 4,000 Hz, which corresponds to passing the signal through a low-pass filter, and the associated "new" synthetic waveform. In this case there are not many long tracks of high frequency components in the range of frequencies removed so there is only a slightly noticeable effect on the waveform. The elimination of the higher frequencies are most apparent when listening to, or "playing" the sound.

Another example is shown in Figure 11 (b) where all frequency tracks less than 20 frames in length are removed, followed by eliminating the frequency range from 3,000 Hz to 4,000 Hz. Figure 11 (a) again shows the original frequency track plot and associated synthetic waveform.

In some cases, it is desired to regenerate the signal using only part of the original signal. Thus, we may need to be able to "cut" a small region in time of the frequency tracks, and then regenerate the signal again. This tool allows the user to indicate the frame range to be cropped. An example is shown in Figure 12 where frames 30 to 40 have been cut, therefore eliminating some of the "silent region" between the two major portions of the

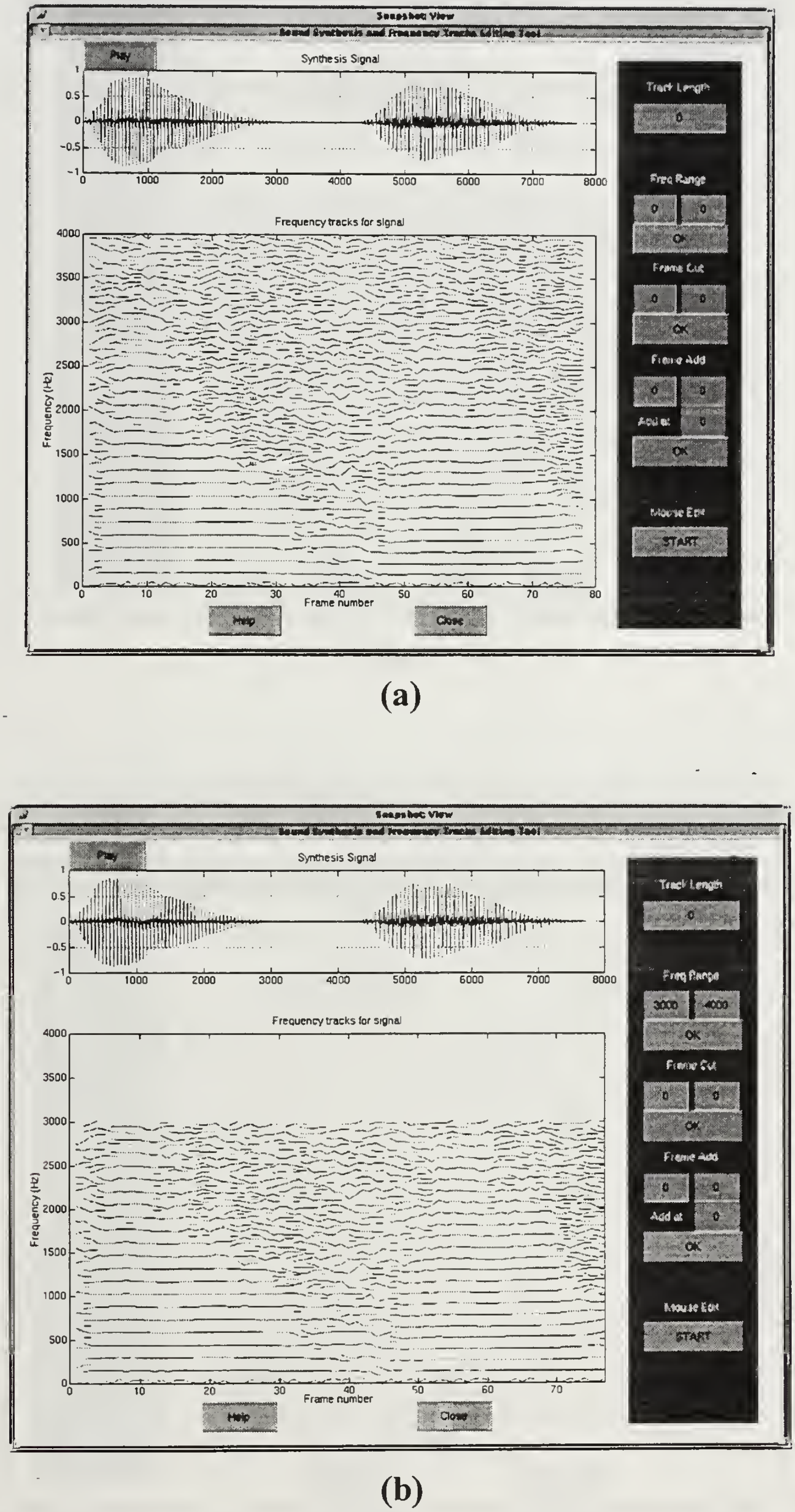


(a)



(b)

Figure 9. Frequency Editing Tool Sample View (a) Original (b) After Frequency Tracks of Length < 20 Frames Have Been Eliminated

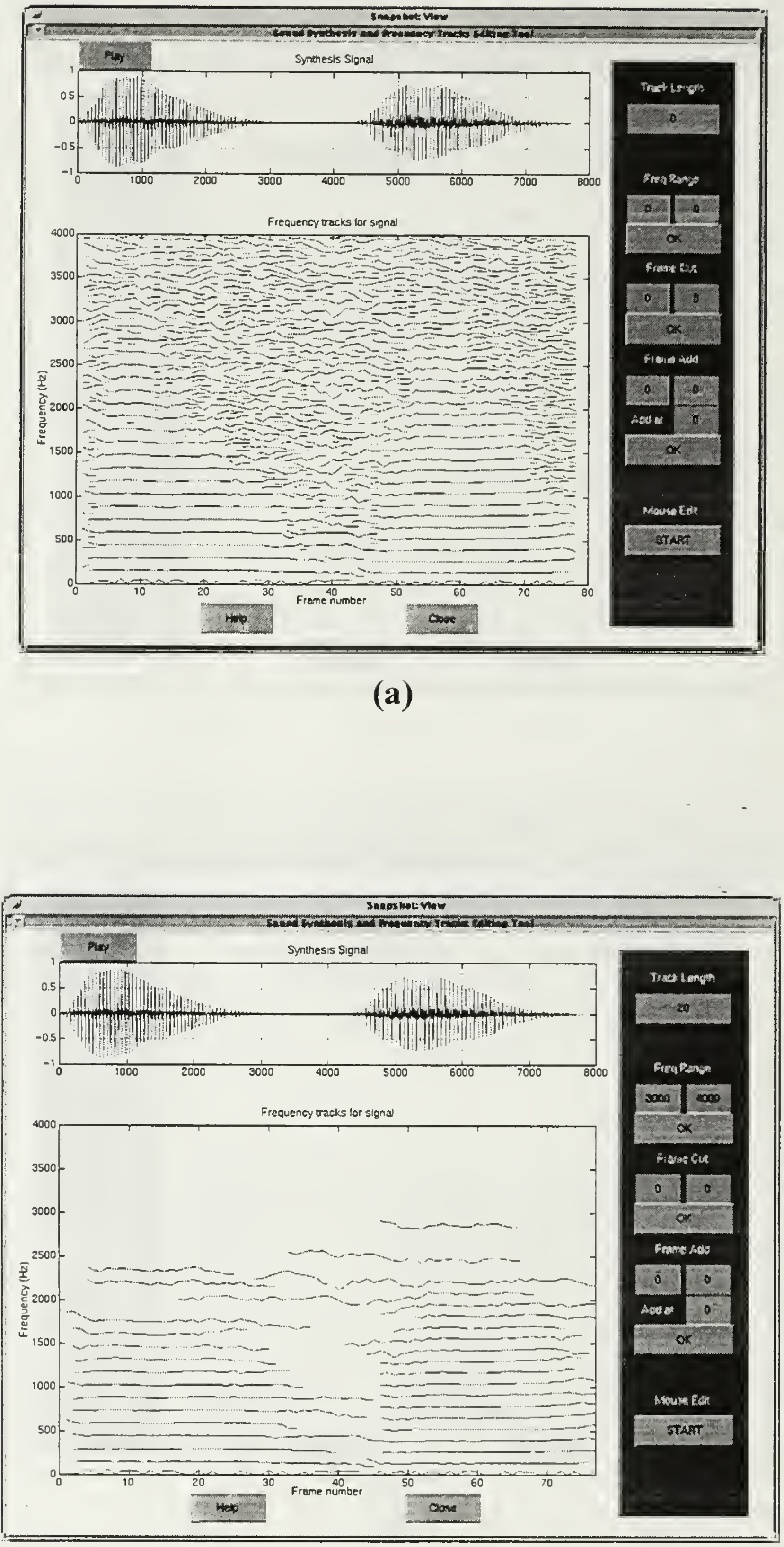


(a)

Freq Range: 3000 4000

(b)

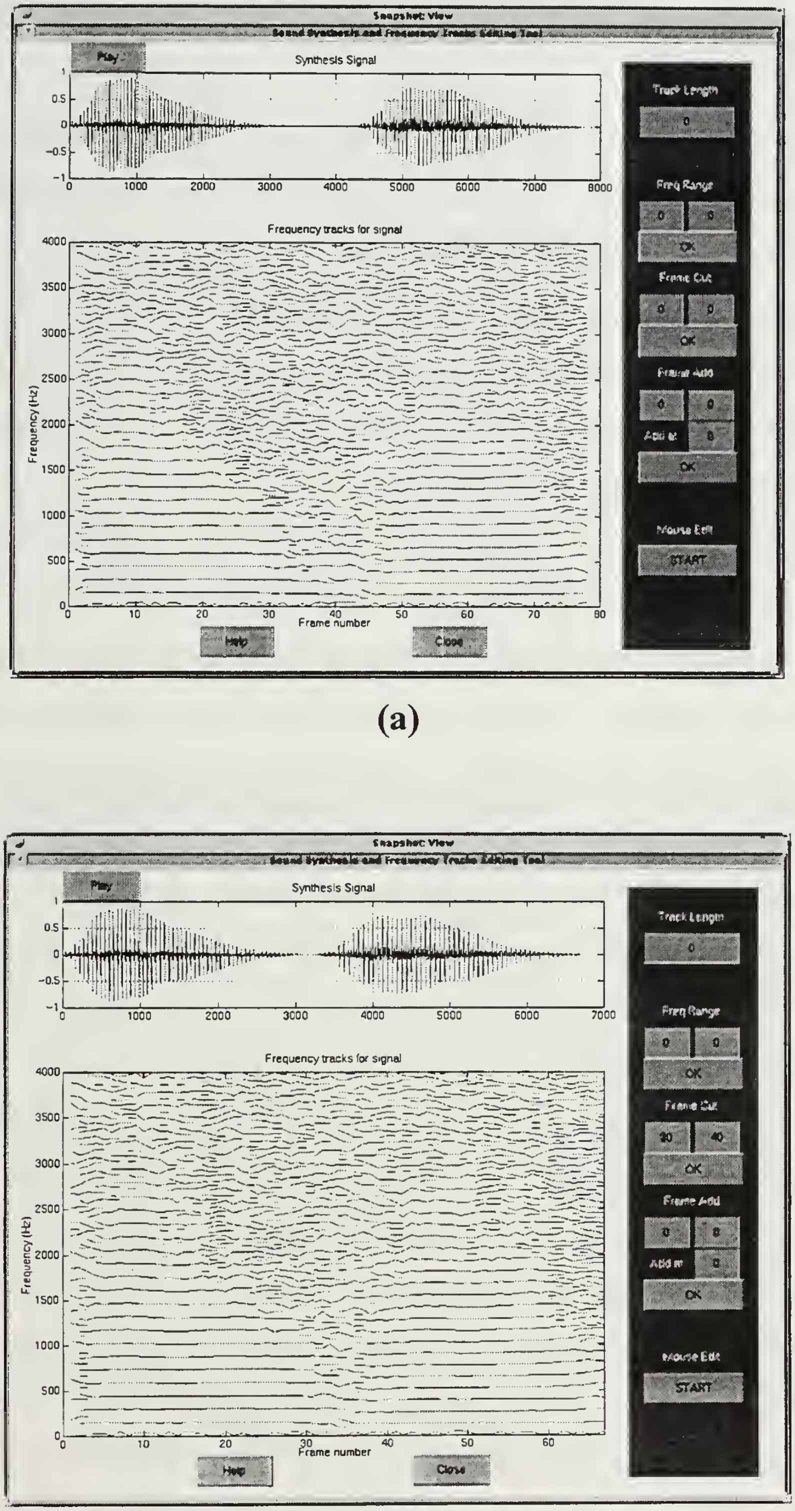
Figure 10. Frequency Editing Tool Sample View (a) Original (b) After Frequency Range From 3,000 Hz to 4,000 Hz Has Been Eliminated



(a)

(b)

Figure 11. Frequency Editing Tool Sample View (a) Original (b) Modified



(a)

Track Length 0

Freq Range 0 0

Frame Cut 0 0

Frame Add 0 0

Add w 0

(b)

Track Length 0

Freq Range 0 0

Frame Cut 30 40

Frame Add 0 0

Add w 0

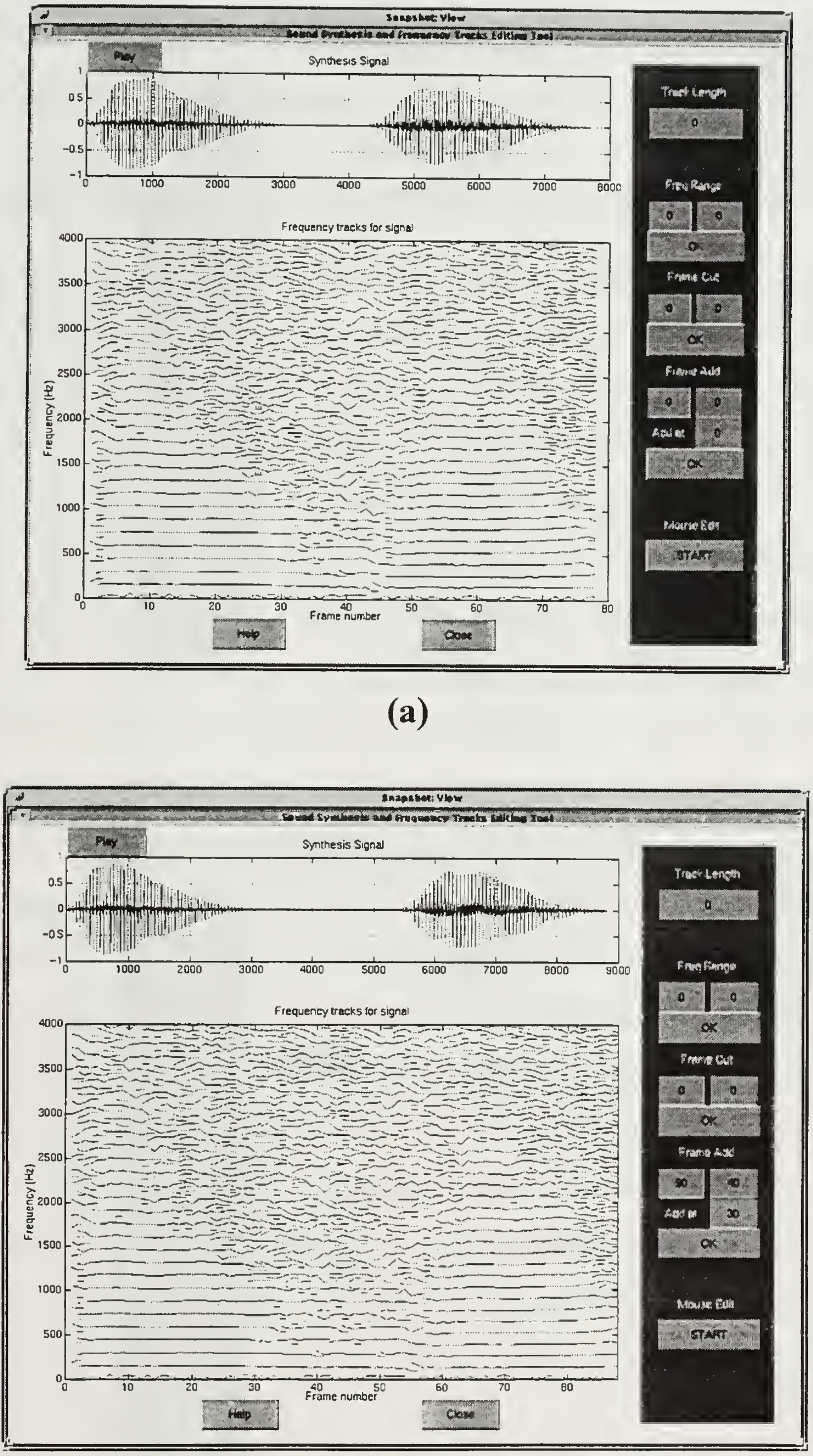
Figure 12. Frequency Editing Tool Sample View (a) Original (b) After Frequency Tracks Frames From 30 to 40 Have Been Cut

sound. The resynthesized waveform is also shown in the top portion of the window in Figure 12 (b).

In other situations, the user may desire to move or repeat a small portion of sound signal at a specific time instant. This can also be done by using the frequency track editing tool. Users are required to input the range of frames to be repeated, and the position in time where the frames are to be inserted. Figure 13 illustrates an example in which frames from 30 to 40 are copied and re-inserted at frame 30. In this case, the result is an increase the "silent region" between the two major portions of the sound. The resynthesized signal is shown in the top portion of the window in Figure 13 (b).

Sometimes, it is desired to remove some *specific* longer frequency tracks after most of the shorter tracks have been removed. Many times this is not possible using the methods that have been previously described. A handy mouse frequency track editing function was developed for this purpose. The user activates this editing function by pressing the "START" push button at the right bottom corner of the window. The cursor changes from an arrow to cross-hairs indicating that the editing function is active. The user then places the cross-hair cursor on a frequency track to be eliminated and "selects" that track with the left mouse button. Every selected frequency track changes from its normal yellow solid color to a red, dotted line. The unaffected frequency components will be saved for the use of generating the new sound signal.

Users are allowed to select as many frequency tracks as they wish to eliminate. The "new modified" synthetic signal is not generated until the user has finished the frequency track selecting step. When the user has finished selecting frequency components he/she presses the right mouse button in the region and then presses the "OK" button to synthesize the waveform. An example of the use of the tool and this function is shown in Figure 14. In Figure 14 (a), all frequency tracks less than 20 frames in length were initially removed. Figure 14 (b) shows the result where three specific additional frequency tracks, one at approximately 3,700 Hz, one at 2,300 Hz and one near 0 Hz, have been eliminated. The



Track Length 0

Freq Range 0 0

Frame Cut 0 0

Frame Add 0 0

Add at 0

(a)

Track Length 0

Freq Range 0 0

Frame Cut 0 0

Frame Add 30 40

Add at 30

(b)

Figure 13. Frequency Editing Tool Sample View (a) Original (b) After Frequency Tracks Frames From 30 to 40 Have Been Repeated at Frame 30

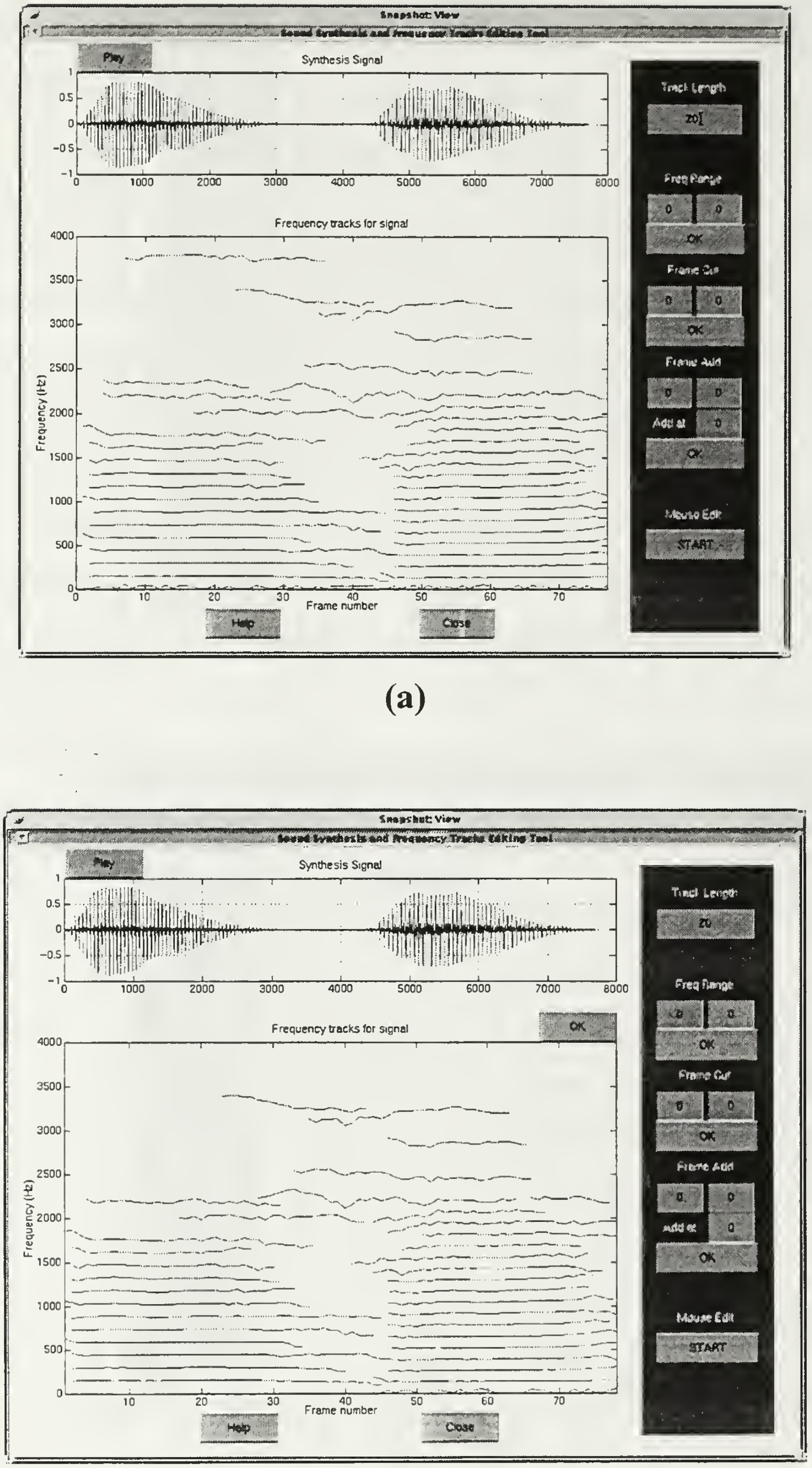


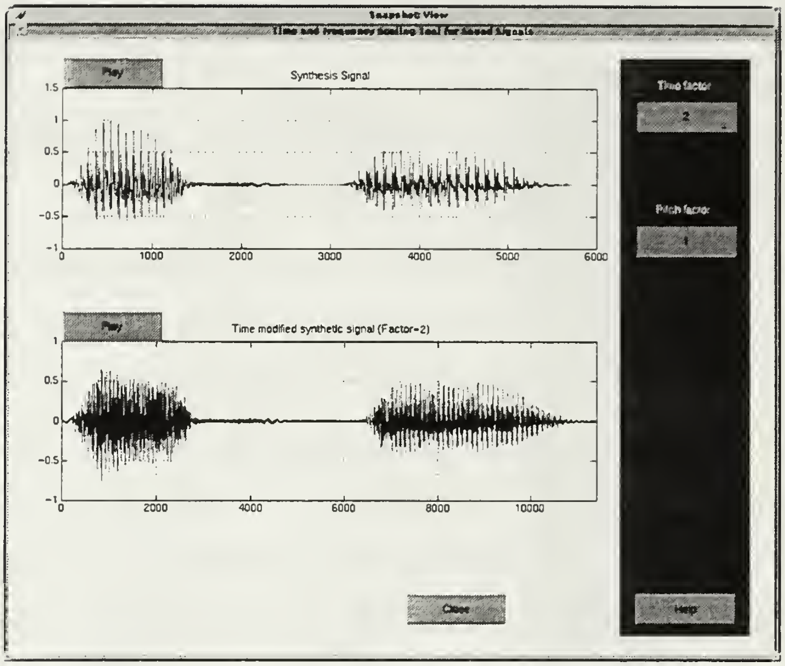
Figure 14. Frequency Editing Tool Sample View (a) Original (b) After Some Frequency Tracks Have Been Eliminated by Mouse Selecting (Dotted Lines)

tracks eliminated are shown as dotted lines and the waveform shown in Figure 14 (b) is the result of synthesis of the removal of these tracks.

C. TIME AND FREQUENCY SCALING INTERACTIVE TOOL

The time and frequency scaling tool provides a means of generating the expansion or compression of a sound signal in the time domain, as well as a change in spectral envelope and pitch contour according to the methods described in Chapter II (B). There are two options offered by this tool. The first is time-scale modification. In the case of time-scale modification, the new sound signal is expanded or compressed depending on the value which the user inputs. The new sound signal is expanded if the value is greater than 1, and is compressed if the value is less than 1. The modified signal is automatically generated right after the user inputs the time-scale factor. An example is illustrated in the top portion of the window as shown in Figure 15 where a segment of male speech is expanded by a factor of 2. In this case, it is found that the rate of articulation has been slowed down while the perceptual quality of the original sound is maintained.

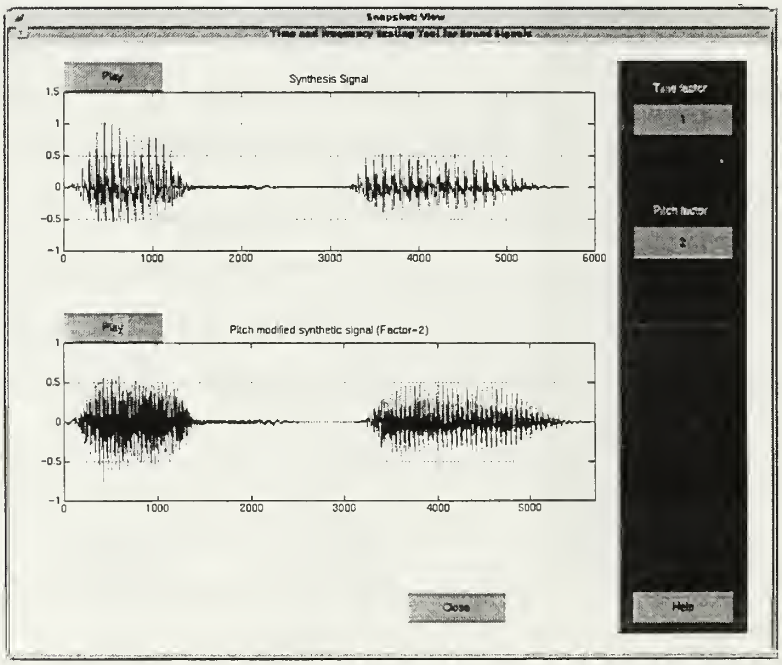
If the user wishes to perform a frequency transformation, then the second option of this tool can be invoked. The user can increase the pitch of the synthesized sound signal by entering a pitch-scaling factor which is greater than 1 or lower the pitch by inputting a value that is less than unity. Both the time- and pitch-scaling factors have been limited to the values in range from 0.1 to 2, since values outside of this range generally produce poor results. An example of frequency scaling by a factor of 2 for male speech is depicted in the bottom portion of the window shown in Figure 16. The resulting speech sounds like a young boy's voice since the pitches of children's voices are higher than those of adults in general.



Time factor: 2

Pitch factor: 1

Figure 15. Time-Scale Expansion of a Segment of Male Speech by a Factor of 2



Time factor: 1

Pitch factor: 2

Figure 16. Frequency Scaling of a Segment of Male Speech by a Factor of 2

IV. CONCLUSIONS

A. DISCUSSION OF RESULTS

In this thesis, a sinusoidal representation model for sound signals by McAulay and Quatieri is described and is used in an analysis/synthesis technique based on the amplitudes, frequencies, and phases of excitation contributions of the sine wave components. In the analysis steps, the data is first sectioned into frames and the discrete Fourier transform (DFT) is applied over each frame. The peaks in the resultant spectrum determine the frequencies of sinusoids to be used in the model and which are “tracked” through successive frames. The amplitudes and phases of the sinusoids are given by the appropriate samples of the DFT corresponding to those peak frequencies.

In the synthesis step, these amplitude and phase functions are applied to the sine-wave generator, which adds all sinusoidal components to produce the synthetic signal output. We have found that this model reproduces the sound very accurately and confirms the claim by the authors that the sound is “perceptually indistinguishable” from the original sound [Ref. 1].

Functional relationships for each of the sine-wave parameters have been developed by the original authors that allow the synthesis system to perform a variety of sound signal transformations, such as time-scale modification and frequency scaling implemented in this thesis [Ref. 4]. These were also found to be effective.

All of the above sound analysis/synthesis sinusoidal representations and transformations were developed as two interactive tools with GUI using MATLAB. In addition, an interactive tool for signal frequency track editing based on the sinusoidal model was also implemented so users can simplify the model and reduce its complexity. Examples of the use of these tools are presented in this thesis.

B. SUGGESTIONS FOR FUTURE STUDY

In the sine-wave-based time and frequency scaling modification system, the modified synthetic waveforms are good in perceptual quality but some structure of the original waveform is lost. A new sine-wave-based speech modification algorithm called the *"Shape Invariant"* technique has been proposed by the original authors which is able to maintain the temporal structure of the original waveform [Ref. 6]. It would be worthwhile to implement this new technique and incorporate it into the existing GUI.

Although this sinusoidal representation model produces very accurate results, it is demanding in terms of computation. Another worthwhile endeavor would be to improve the computational performance of the sine-wave-based modification system. MATLAB is poor in executing "Loop Iteration" code because it is an interpreted language. Unfortunately, the implementation of the modification system uses several "loops" which makes the execution time quite long. Rewriting the code to improve the computational efficiency would be very desirable so that the user can see the results more quickly. Perhaps compiling the code with the MATLAB to C (or C++) compiler would result in faster execution.

APPENDIX

This appendix includes the sound signal examples and main programs described in this thesis. Several music and speech signals were used in the analysis/synthesis, frequency track editing, and transformation experiments, only two of them are presented in this thesis, however. They are:

- The speech phrase "baseball" from a male speaker (file: obase.mat), and
- Two notes (file: tbhi2.mat) excerpted from a four-note trombone passage (file: tb_hi.au).

The program is written entirely in MATLAB and makes extensive use of Graphical User Interface (GUI) features from MATLAB. In addition, the MATLAB Signal Processing Toolbox is required to run these programs.

The programs are divided into three parts, the sound analysis/synthesis, frequency track editing, and sound transformation (including time-scale modification and frequency scaling). They correspond to Chapter III (A), (B), and (C), respectively.

A. SOUND ANALYSIS/SYNTHESIS FUNCTIONS

[cand, mag, phas, sig, par, synt] = gsinwave(signal);

This is the main program which calls other associated MATLAB functions in order to perform the sound analysis/synthesis. The user invokes this program. This function calls guisynth.m which implements all GUI actions and calls the two other functions analy.m and synth.m which perform the anlysis and synthesis, respectively. The input to gsinwave.m is the original sound signal and the outputs are the synthesized signal (synt) and a set of variables which are generated from the analy.m and synth.m functions. These other variables are described below.

1. Analysis Function

```
[cand, mag, phas, sig, par] = analy(signal, sam_fs, N, n, peak_thr, mat_win, fft_size);
```

			Default Values
Inputs	signal	original sound signal	
	sam_fs	sampling frequency	8000 Hz
	N	windowed frame length	200 points
	n	overlap width	100 points
	peak_thr	peak picking threshold	80 dB
	mat_win	frequency matching interval	10
	fft_size	DFT points	1024
Outputs	cand	sinusoids peak candidate matrix	
	mag	magnitude matrix of DFT	
	phas	phase matrix of DFT	
	sig	windowed original signal	
	par	parameters used by other functions	

The output variables listed above can be described in more detail as follows.

- cand

  This matrix contains the "matched" (i.e., after applying the "birth" and "death" process and "frame-to-frame" peak matching procedure) peak frequencies information of sinusoids which are extracted from the DFT spectrum. The number of rows of this matrix is equal to one half of the DFT points, and the number of columns depends on both the windowed frame length and length of the sound signal.

  A cand matrix of the sound tbhi2.mat was generated by using the default input values mentioned earlier in the analysis function. The size of this cand matrix is 512 by 77. A small matrix corresponding to rows 36 to 60 and columns 1 to 10 was excerpted from the cand matrix as an example. The location and value of each element indicate the frequency and the "matched" position on the following frame, respectively.

  For instance, the element (37,2) of the original cand matrix corresponds to the frequency value 289 Hz. The value "38" in this position of the matrix indicates that the element (38,3) is the next matched position ("38" corresponds to the frequency value 297 Hz). Also, since there is no value "37" in column 1 of this matrix, this frequency track is considered to be "born" in frame 1. Let us examine another element (53,2) of this matrix. Its value "53" indicates that the element (53,3) would possible be a nonzero value so that this frequency track can continue. However, since the value of element (53,3) is zero, the track "dies" at this point.

	Column (Frame)									
Row	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
36	0	0	0	0	0	0	0	0	0	0
37	0	38	0	0	0	0	0	0	0	0
38	0	0	38	38	38	39	0	39	0	38
39	0	0	0	0	0	0	38	0	38	0
40	0	0	0	0	0	0	0	0	0	0
41	0	0	0	0	0	0	0	0	0	0
42	0	0	0	0	0	0	0	0	0	0
43	0	0	0	0	0	0	0	0	0	0
44	0	0	0	0	0	0	0	0	0	0
45	0	0	0	0	0	0	0	0	0	0
46	0	0	0	0	0	0	0	0	0	0
47	0	0	0	0	0	0	0	0	0	0
48	0	0	0	0	0	0	0	0	0	0
49	0	0	0	0	0	0	0	0	0	0
50	0	0	0	0	0	0	0	0	0	0
51	0	0	0	0	0	0	0	0	0	0
52	0	0	0	0	0	0	0	0	0	0
53	53	53	0	0	0	0	0	0	0	0
54	0	0	0	0	0	0	0	0	0	0
55	0	0	0	0	0	0	0	0	0	0
56	0	0	0	57	0	0	0	0	0	0
57	0	0	56	0	57	57	57	57	57	57
58	0	0	0	0	0	0	0	0	0	0
59	0	0	0	0	0	0	0	0	0	0
60	0	57	0	0	0	0	0	0	0	0

- mag

This is the matrix which contains the magnitudes of DFT. The number of rows of this matrix is equal to one half of the DFT points, and the number of columns depends on the sound signal.

- phas

This is the matrix which contains the phases of DFT. It has the same size as the mag matrix.

- sig

  This is a windowed version of the original sound signal.

- par

  This vector is composed of the essential parameters provided by the user including the sampling frequency, windowed frame length, overlap width, and the number of DFT points which were used in the synthesis and other programs.

2. Synthesis Function

```
[sig, synt] = synth(cand, mag, phas, sig, par);
```

Inputs	All input parameters are generated by the analysis function. (see above description)	
Outputs	sig	windowed signal
	synt	synthesized signal

B. FREQUENCY TRACK EDITING FUNCTIONS

```
[ncand, nmag, nphas, nsig, npar, nsynt] = guifreq(cand, mag, phas, sig, par, synt);
```

This is the main program which needs to be invoked if the user wishes to perform the frequency track editing. This function calls the associated function guiedit.m which performs all GUI and "frequency track editing" operations. The inputs are the same variables used in the analysis and synthesis functions mentioned earlier and the outputs are modified versions of these variables. For example, ncand is the resulting candidate matrix after the original candidate matrix cand has been "edited." The functions called by guiedit.m are described below:

edit.m	eliminates short frequency tracks
zero.m	eliminates a specific range of frequencies
cut.m	deletes a small portion of signal
paste.m	cuts a small portion of signal and pastes it at a specific time instant
mousedit.m	implements the mouse editing capability

C. SOUND TRANSFORMATION FUNCTIONS

```
[newsynt] = guimod(cand, mag, phas, sig, par, synt);
```

This is the main program which calls the associated MATLAB function guiscale.m in order to perform the sound transformations, including time-scale modification and frequency scaling. The inputs are from one of the previous two programs, namely, the sound analysis/synthesis or frequency track editing programs, and the output is the modified waveform. The function guiscale.m includes all GUI and calls function modsynt.m which is required to perform sound transformations.

The user can also get on-line help by typing "help func_name" in the MATLAB workspace to look at the description about how to use the above functions. Alternatively, one can simply activate these interactive GUI tools and press the "Help" button to get more information.

LIST OF REFERENCES

1. McAulay, Robert J., and Thomas F. Quatieri, "Speech Analysis/Synthesis Based on a Sinusoidal Representation," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. ASSP-34, No. 4, pp. 744-754, August 1986.

2. Serra, Xavier, "A System for Sound Analysis/Transformation/Synthesis Based on a Deterministic plus Stochastic Decomposition," Report No. STAN-M-58, CCRMA, Department of Music, Stanford University, October 1989.

3. Victory, Charles W., "Comparison of Signal Processing Modeling Methods for Passive Sonar Data," Master Thesis, Naval Postgraduate School, March 1993.

4. Quatieri, T. F., and R. J. McAulay, "Speech Transformation Based on a Sinusoidal Representation," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. ASSP-34, No. 6, pp. 1449-1464, December 1986.

5. *Building a Graphical User Interface* by the MathWorks Inc., June 1993.

6. Quatieri, Thomas F., and Robert J. McAulay, "Shape Invariant Time - Scale and Pitch Modification of Speech," *IEEE Transactions on Signal Processing*, Vol. 40, No. 3, pp. 497-510, March 1992.

BIBLIOGRAPHY

Brown, Dennis, W., "SPC Toolbox: A MATLAB Based Software Package for Signal Analysis," Master Thesis, Naval Postgraduate School, September 1995.

Deller, John R., Jr., John G. Proakis, John H. L. Hansen, *Discrete-Time Processing of Speech Signals*, Englewood Cliffs, New Jersey: Prentice Hall, 1987.

Haykin, Simon, *Communication Systems*, New York: John Wiley & Sons, Inc., 1994.

Portnoff, M. R., "Time-Scale Modification of Speech Based on Short-Time Fourier Analysis," *IEEE Transactions on Acoustics, Speech, and Signal Processing,* Vol. ASSP-30, No. 3, pp. 374-390, June 1981.

Therrien, Charles W., *Discrete Random Signals and Statistical Signal Processing*, Englewood Cliffs, New Jersey: Prentice Hall, 1992.

INITIAL DISTRIBUTION LIST

No. Copies

1. Defense Technical Information Center 2
8725 John J. Kingman Rd., STE 0944
Ft. Belvoir, VA 22060-6218

2. Dudley Knox Library 2
Naval Postgraduate School
411 Dyer Rd.
Monterey, California 93943-5101

3. Professor Charles W. Therrien 4
Code EC/TI
Naval Postgraduate School
Monterey, California 93943

4. Professor Roberto Cristi 1
Code EC/CX
Naval Postgraduate School
Monterey, California 93943

5. Library of Chung Cheng Institute of Technology 1
P.O. Box 90047
Ta-hsi, Taoyuan
TAIWAN, R.O.C.

6. Major Hsiao-Tseng Lin 1
SGC #2552 NPS
Monterey, California 93943

7. LT Ming-Fei Chuang 2
No. 34, Lane 152, Sec. 3 Yuanlin Rd.
Ta-hsi, Taoyuan
TAIWAN, R.O.C.